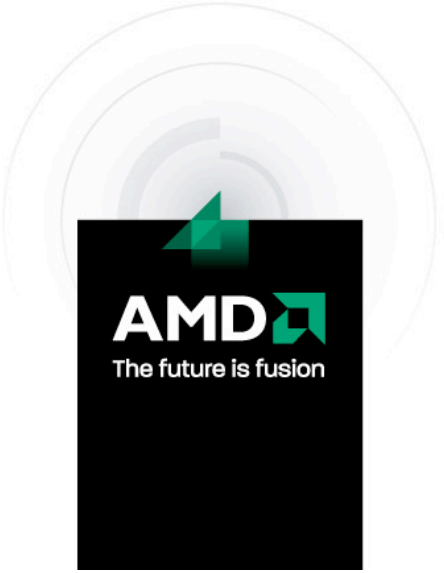
AMD Opteron™ Processors



Outline

- Hardware Overview
- Software Overview
- Questions

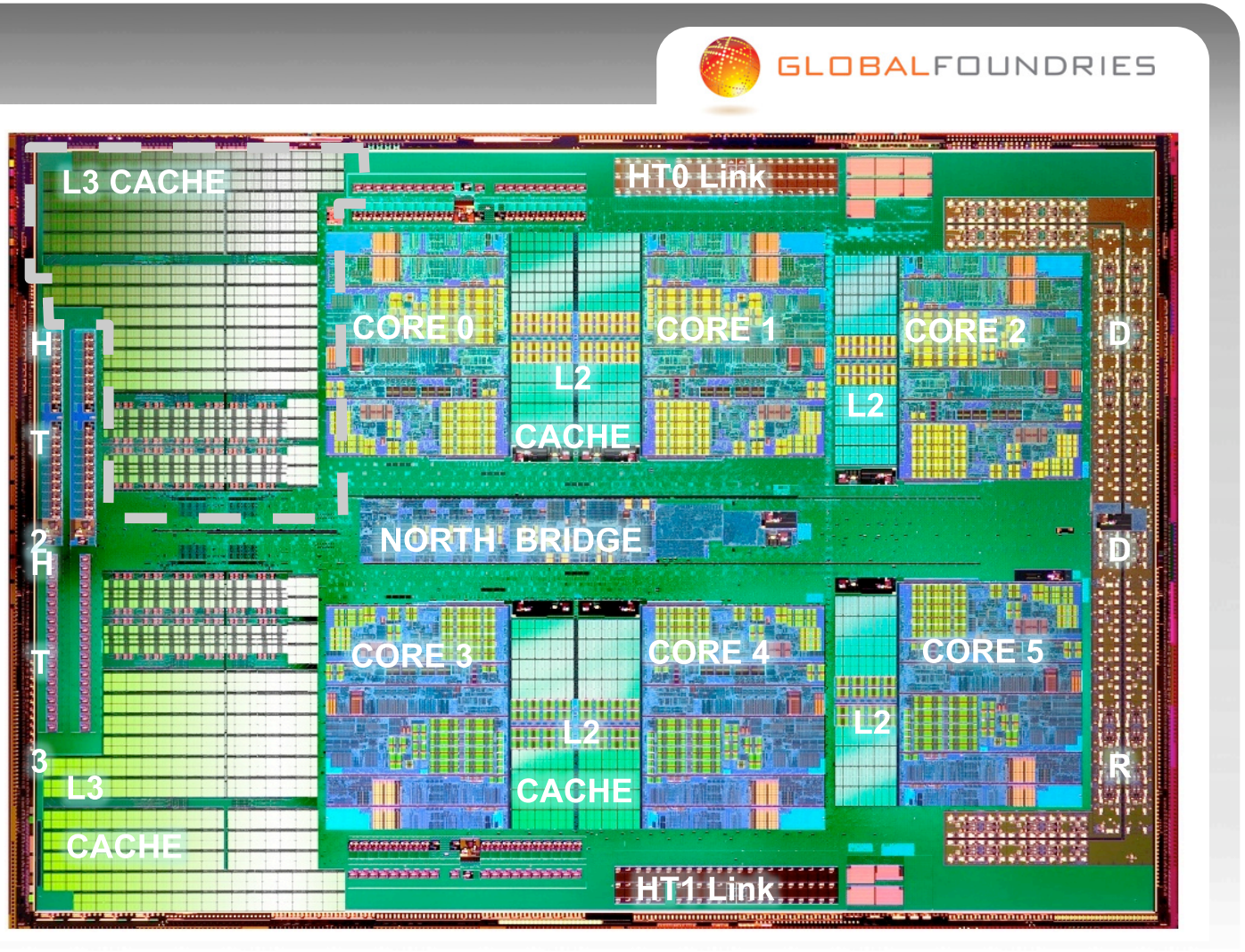
Hardware

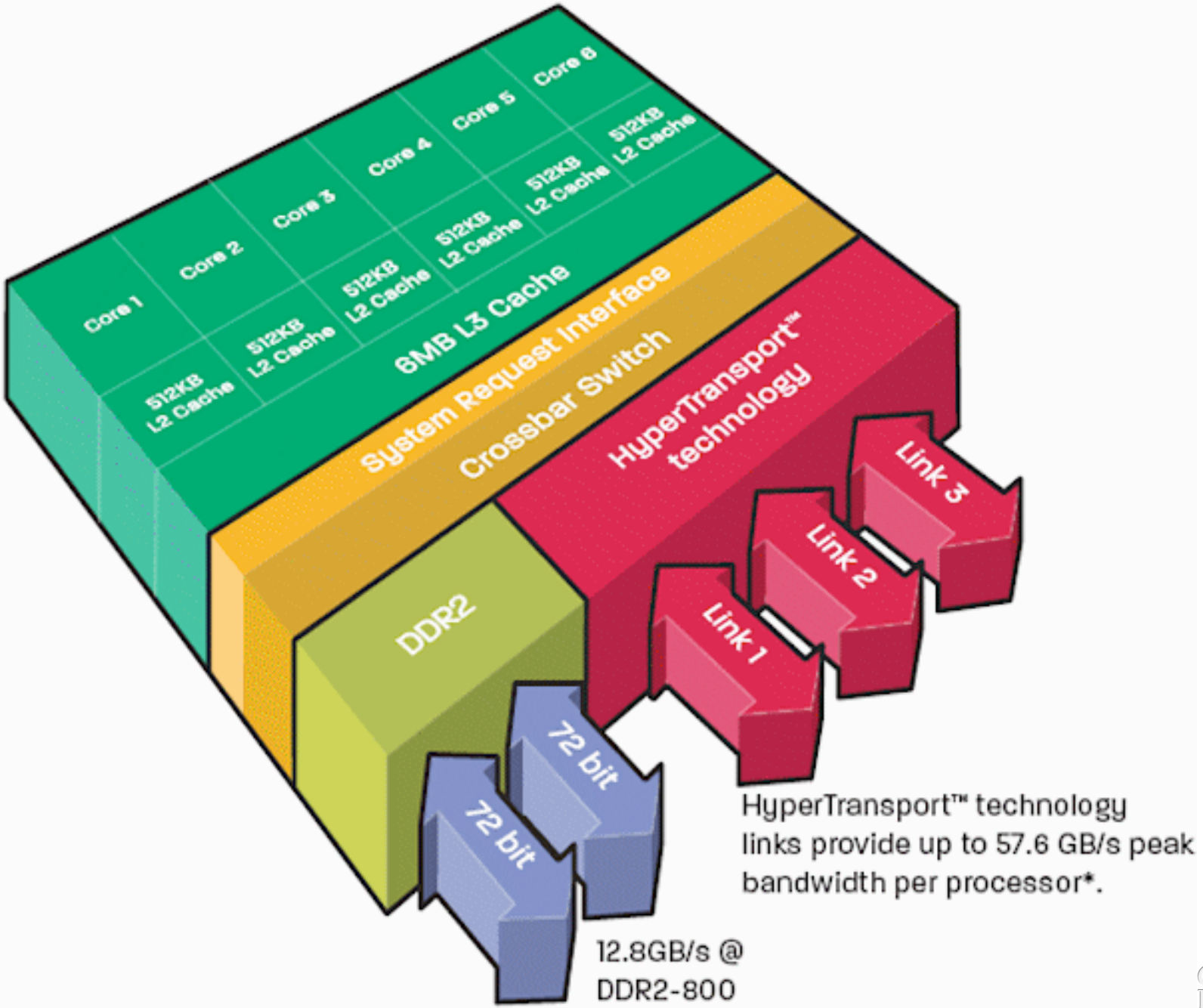
x86 64-bit Architecture Evolution

	2003	2005	2007	2008	2009	2010
	AMD Opteron™	AMD Opteron™	"Barcelona"	"Shanghai"	"Istanbul"	"Magny-Cours"
Mfg. Process	90nm SOI	90nm SOI	65nm SOI	45nm SOI	45nm SOI	45nm SOI
CPU Core	K8	K8	Greyhound	Greyhound+	Greyhound+	Greyhound+
L2/L3	1MB/0	1MB/0	512kB/2MB	512kB/6MB	512kB/6MB	512kB/12MB
Hyper Transport™Technology	3x 1.6GT/s	3x 1.6GT/s	3x 2GT/s	3x 4.0GT/s	3x 4.8GT/s	4x 6.4GT/s
Memory	2x DDR1 300	2x DDR1 400	2x DDR2 667	2x DDR2 800	2x DDR2 1066	4x DDR3 1333

Max Power Budget Remains Consistent

“Istanbul” Silicon





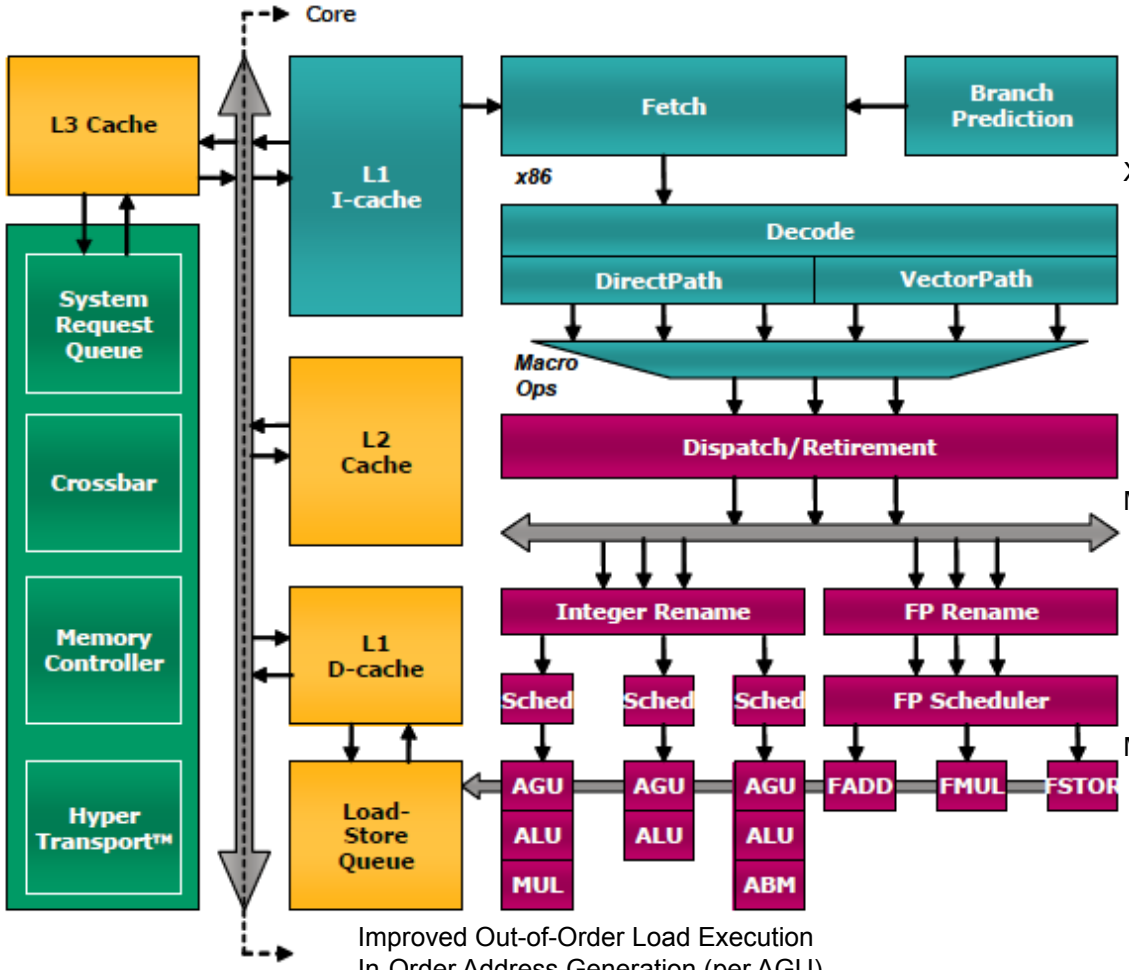
Shanghai to Istanbul

- 6 cores (~1.5X flops)
  - Same per core L1 & L2
  - Same shared L3
  - NB & Xbar upgrades (going from 4 to 6 cores)
- HT Assist – provides 3 probe scenarios
  - No probe needed
  - Directed probe
  - Broadcast probe
- Memory BW and latency improvement
  - Amount depends on platform and configuration
- Socket Compatibility

Core Micro Architecture

FastPath? Macro-Ops? Micro-Ops?

Reference : Software Optimization Guide for AMD Family 10h Processors, Pub. #40546, Rev. 3.10 Feb 2009



X86 Insts.

Macro-Ops

Micro-Ops

Improved Out-of-Order Load Execution
In-Order Address Generation (per AGU)
Store-to-Load Forwarding Support

Notes / Considerations

Avoid having more than 2 or 3 branches per 16B of instructions.

Three Decode Categories (FastPath also called DirectPath)
- DirectPath Single - *best*
- DirectPath Double - *better*
- VectorPath (microcode) - *good*

Macro-Ops tracked ReOrder Buffer (ROB).
- 72 entries (3 wide x 24 deep)
- In-order dispatch, retirement

Micro-Ops issue from Sched to Execution Units
- "Sched" aka "Reserv. Station"
- Out-of-order issue
- FP scheduler shared across units
- INT Schedulers are "per unit"

Six-Core AMD Opteron™ Processor

Performance
- Six-Core AMD Opteron™ Processor
  - 6M Shared L3 Cache
  - North Bridge enhancements (PF + prefetch)
  - 45nm Process Technology
- DDR2-800 Memory
- HyperTransport-3 @ 4.8 GT/sec

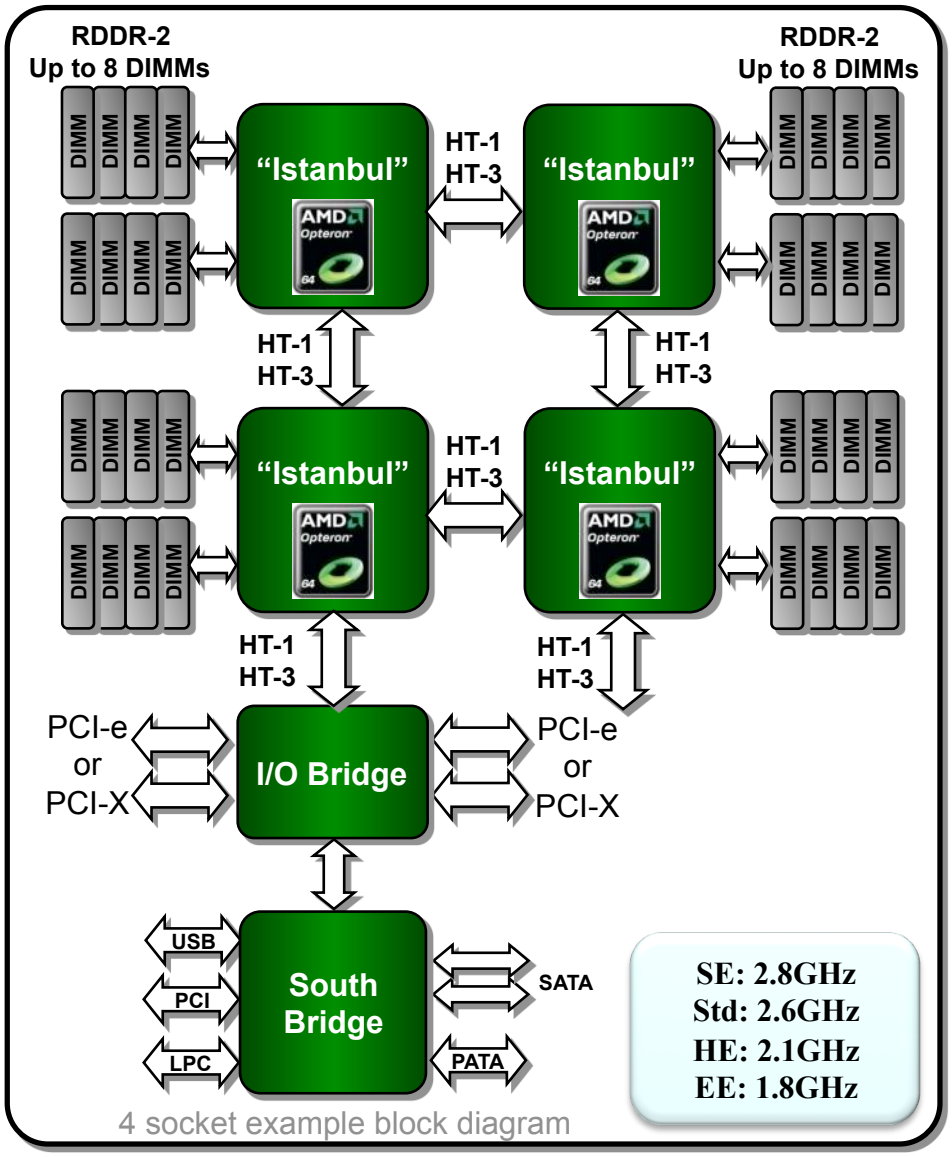
Reliability/Availability
- L3 Cache Index Disable
- HyperTransport Retry (HT-3 Mode)
- x8 ECC (Supports x4 Chipkill in unganged mode)

Virtualization
- AMD-V™ with Rapid Virtualization Indexing

Manageability
- APML Management Link*

Scalability
- 48-bit Physical Addressing (256TB)
- HT Assist (Cache Probe Filter)

Continued Platform Compatibility
- Nvidia/Broadcom-based F/1207 platforms



4 socket example block diagram

Six-Core AMD Opteron™ Processor

Performance
- Six-Core AMD Opteron™ Processor
  - 6M Shared L3 Cache
  - North Bridge enhancements (PF + prefetch)
  - 45nm Process Technology
- DDR2-800 Memory
- HyperTransport-3 @ 4.8 GT/sec

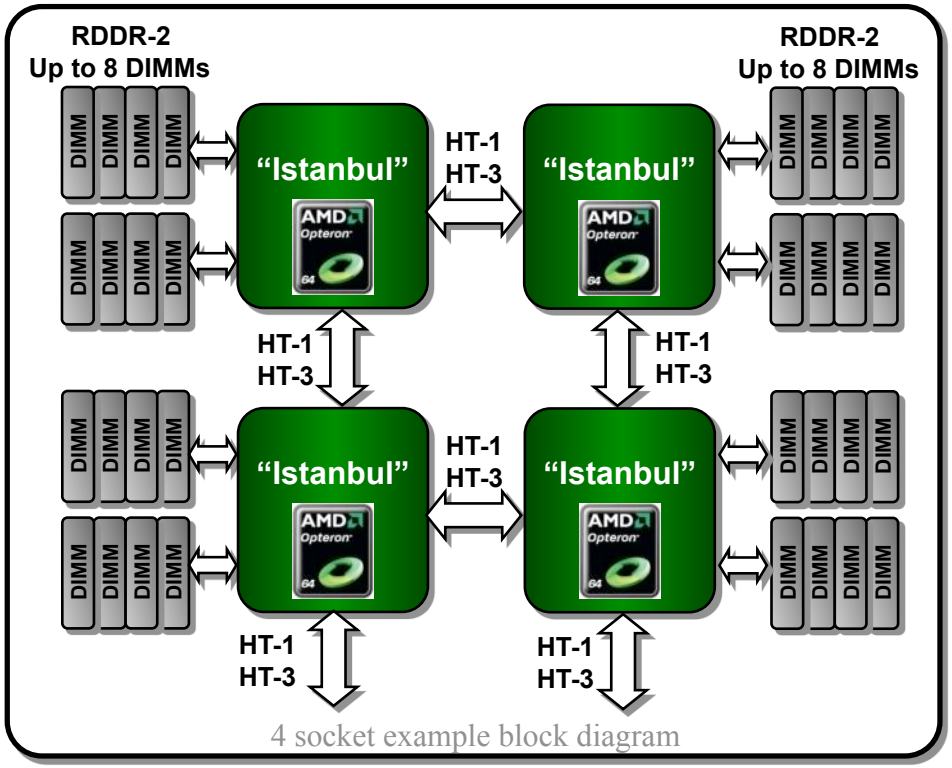
Reliability/Availability
- L3 Cache Index Disable
- HyperTransport Retry (HT-3 Mode)
- x8 ECC (Supports x4 Chipkill in unganged mode)

Virtualization
- AMD-V™ with Rapid Virtualization Indexing

Manageability
- APML Management Link*

Scalability
- 48-bit Physical Addressing (256TB)
- **HT Assist (Cache Probe Filter)**

Continued Platform Compatibility
- Nvidia/Broadcom-based F/1207 platforms



4 socket example block diagram

	STREAM Bandwidth (GB/s)*		
Product, Freq, Dram	2S	4S	8S
"Barcelona," 2.3/2.0, RDDR2-667	17.2	20.5	
"Shanghai," 2.7/2.2, RDDR2-800	21.4	24	
"Istanbul," 2.4/2.2, RDDR2-800	22	**42**	**81.5**

Cache Hierarchy

Dedicated L1 cache

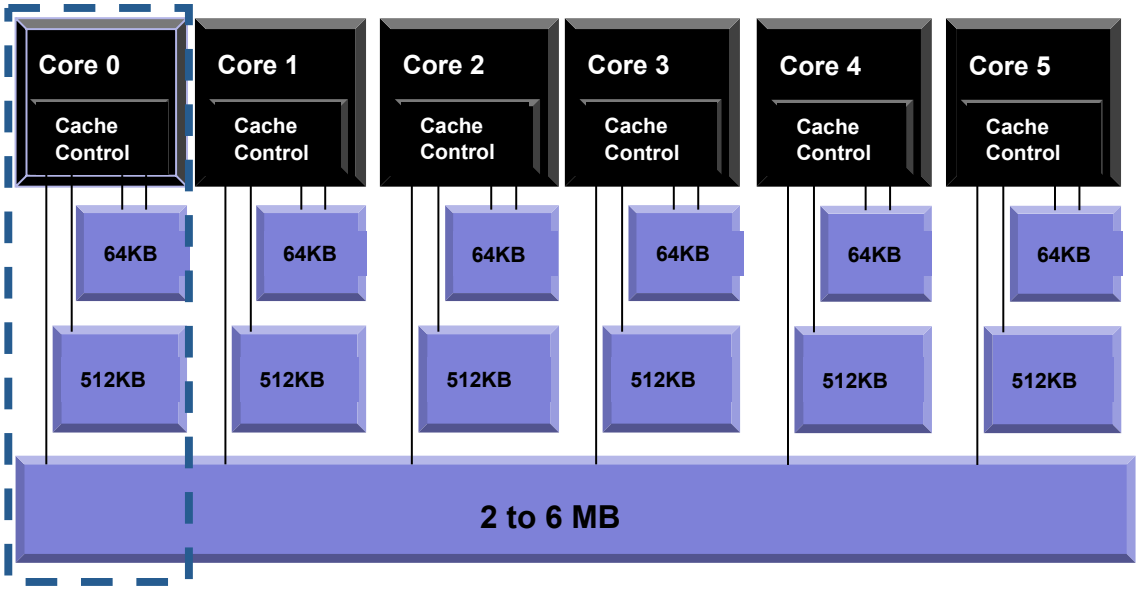
- 2 way associativity.
- 8 banks.
- 2 128-bit loads per cycle.

Dedicated L2 cache

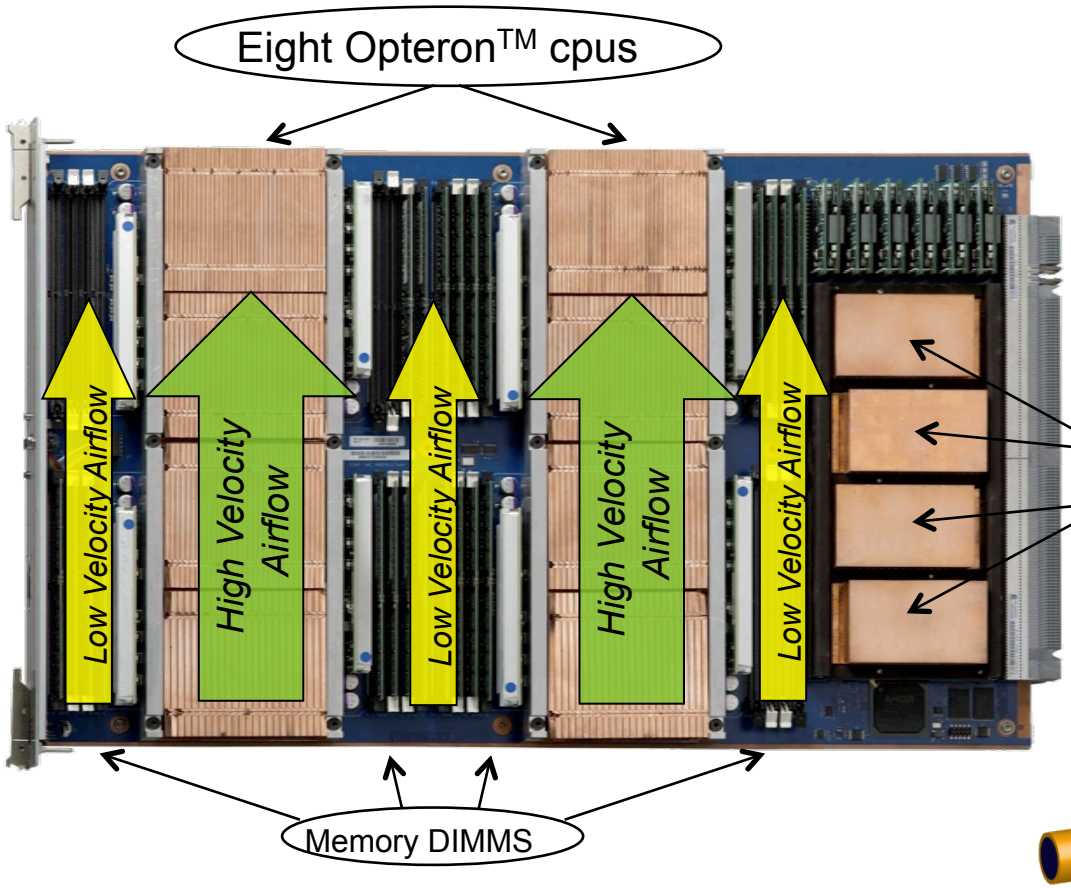
- 16 way associativity.

Shared L3 cache

- 32 way (Barcelona), 48 way (Shanghai and Istanbul) associativity.
- fills from L3 leave likely shared lines in L3.
- sharing aware replacement policy.

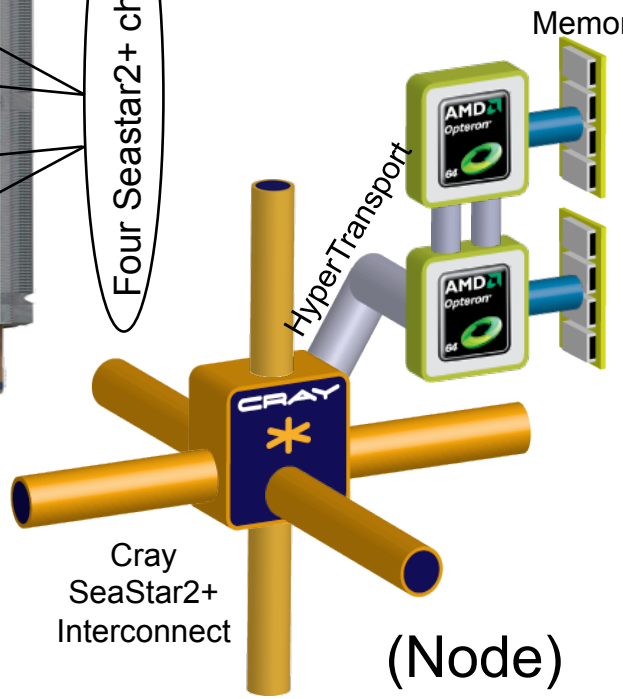


Cray XT5 Blade and Compute Node



Four Compute Nodes per Blade
(2 cpus per node)

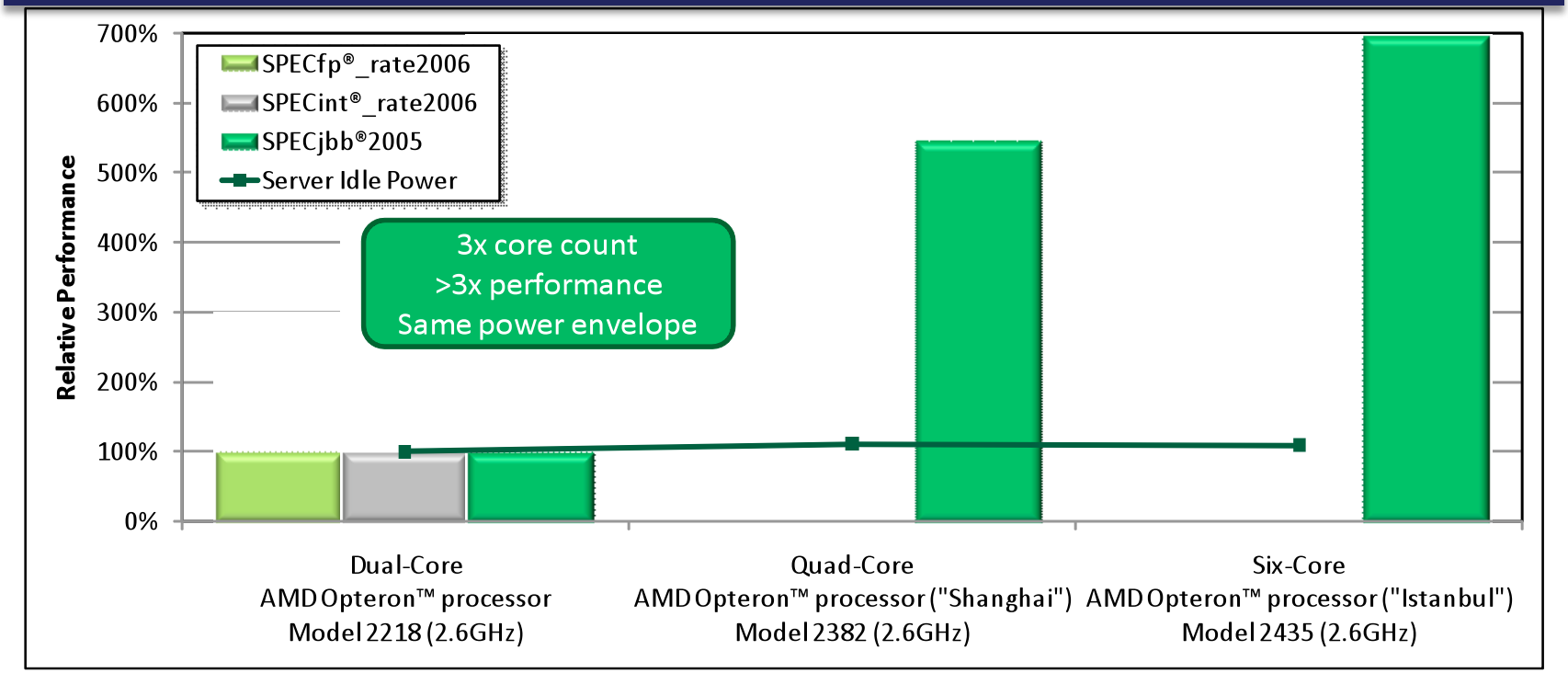
Seastar2+ chips form 3-D torus interconnect



(Node)

Significantly Higher Performance in Same Power Envelope

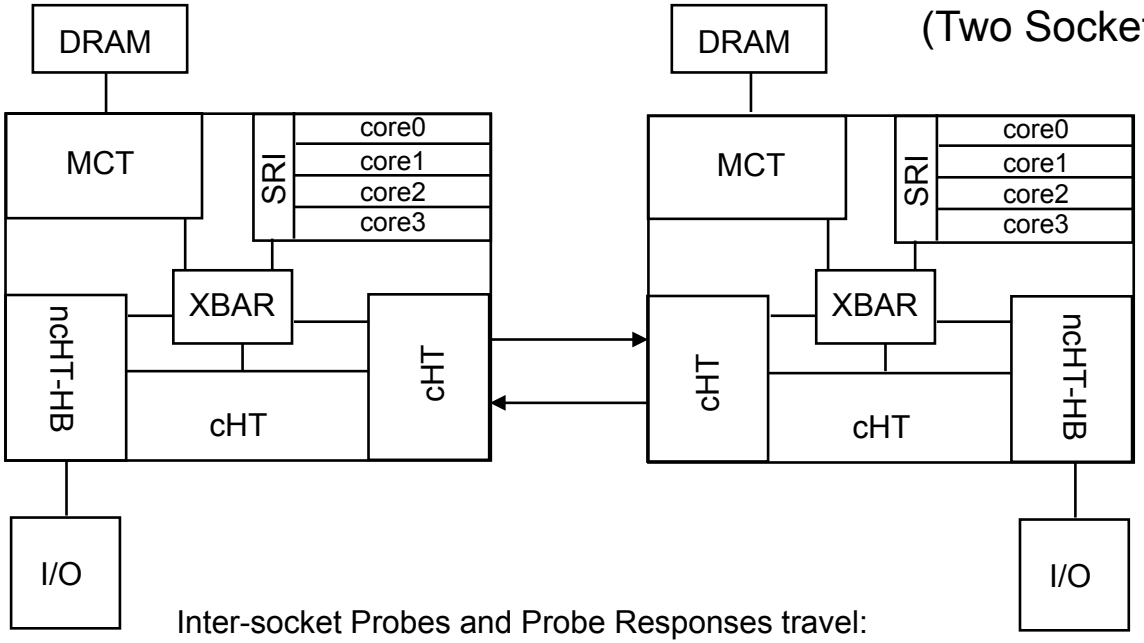
Two-socket servers using Six-Core AMD Opteron™ processors significantly outperform two-socket servers using Dual-Core and Quad-Core AMD Opteron™ processors without consuming significantly more power



SPEC, SPECfp, SPECint, and SPECjbb are registered trademarks of the Standard Performance Evaluation Corporation. The performance results for Six-Core AMD Opteron™ processor Model 2435 and the SPECjbb result for Quad-Core AMD Opteron™ processor Model 2382 are based upon data submitted to Standard Performance Evaluation Corporation as of May 21, 2009. The other performance results stated below reflect results published on http://www.spec.org/ as of May 21, 2009. The server idle power results are based on measurements of server active idle power for 60 seconds at AMD performance labs as of May 21, 2009. The comparison presented below is based on the best performing two-socket servers using AMD Opteron™ processor Models 2218, 2382, and 2435. For the latest results, visit http://www.spec.org/. Please see backup slides for configuration information.

HT Assist Feature (Probe Filters)

Multi-Socket System Overview



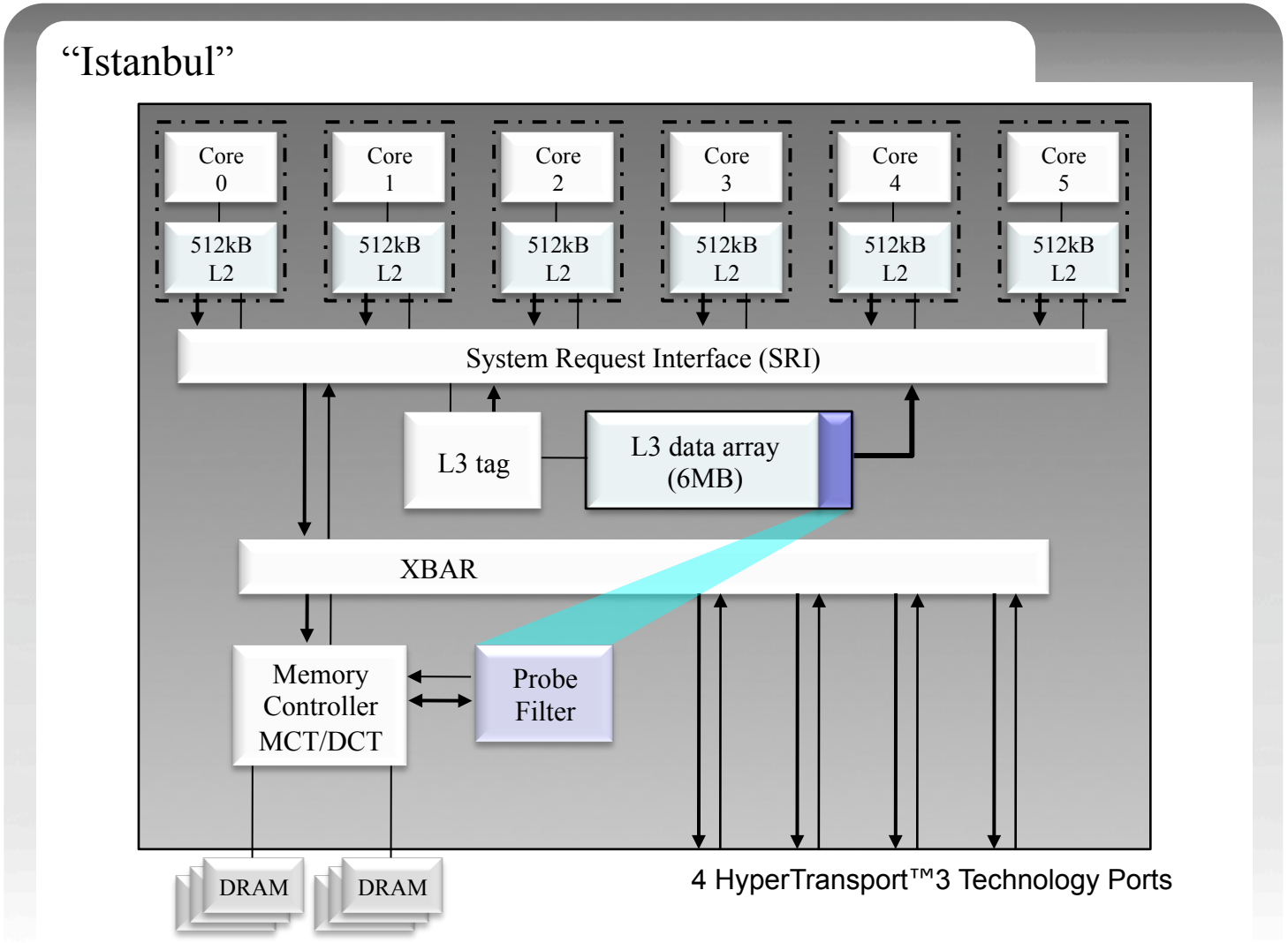
(Two Socket System)

Inter-socket Probes and Probe Responses travel:
SRI -> XBAR -> cHT -> cHT -> XBAR -> SRI

Probes Requests initiate at home memory node, but return directly to node making initial memory request.

key:
- cHT = coherent HyperTransport
- ncHT = non-coherent HyperTransport
- XBAR = crossbar switch
- SRI = system request interface (memory access, cache probes, etc.)
- MCT = memory controller
- HB = host bridge (e.g. HT to PCI, SeaStar, etc.)

Istanbul Block Diagram



HT Assist and Memory Latency

- With "old" broadcast coherence protocol, the latency of a memory access is the longer of 2 paths:
  - time it takes to return data from DRAM and
  - the time it takes to probe all caches
- With HT Assist, local memory latency is significantly reduced as it is not necessary to probe caches on other nodes.
- Several server workloads naturally have ~100% local accesses
  - SPECint®, SPECfp®
  - VMMARK™ typically run with 1 VM per core
  - SPECpower_ssj® with 1 JVM per core
  - STREAM

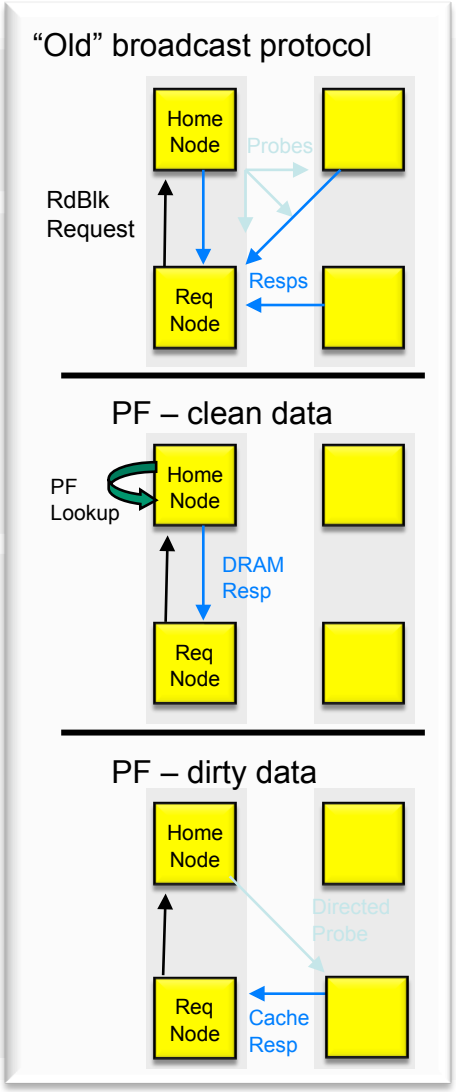
Probe Filter amplifies benefit of any NUMA optimizations in OS/application which make memory accesses local

HyperTransport™ Technology HT Assist (Probe Filter)

- Key enabling technology on “Istanbul” and “Magny-Cours”

- HT Assist is a sparse directory cache
  - Associated with the memory controller on the home node
  - Tracks all lines cached in the system from the home node

- Eliminates most probe broadcasts (see diagram)
  - Lowers latency
    - local accesses get local DRAM latency, no need to wait for probe responses
    - less queuing delay due to lower HT traffic overhead
  - Increases system bandwidth by reducing probe traffic



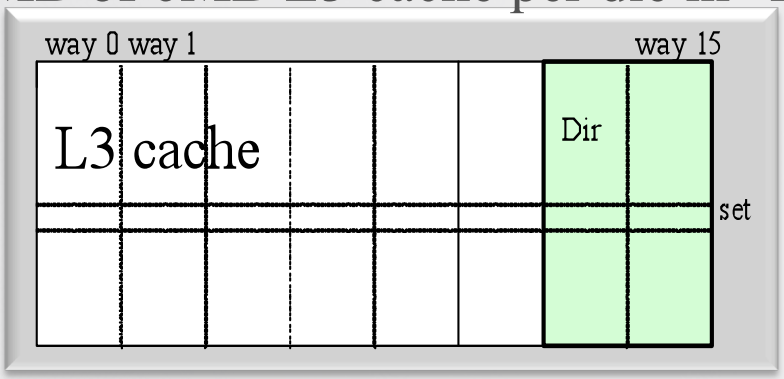
Cache Coherence Protocol

- Track lines in M, E, O or S state in probe filter
- PF is fully inclusive of all cached data in system
  - if a line is cached, then a PF entry must exist.
- Presence of probe filter entry says line in M, E, O or S state
- Absence of probe filter entry says line is uncached
- New messages
  - Directed probe on probe filter hit
  - Replacement notification E ->I (clean VicBlk)

Where Do We Put the HT Assist Probe Filter?

- **Q:** Where do we store probe filter entries without adding a large on-chip probe filter RAM which is not used in a 1P desktop system?

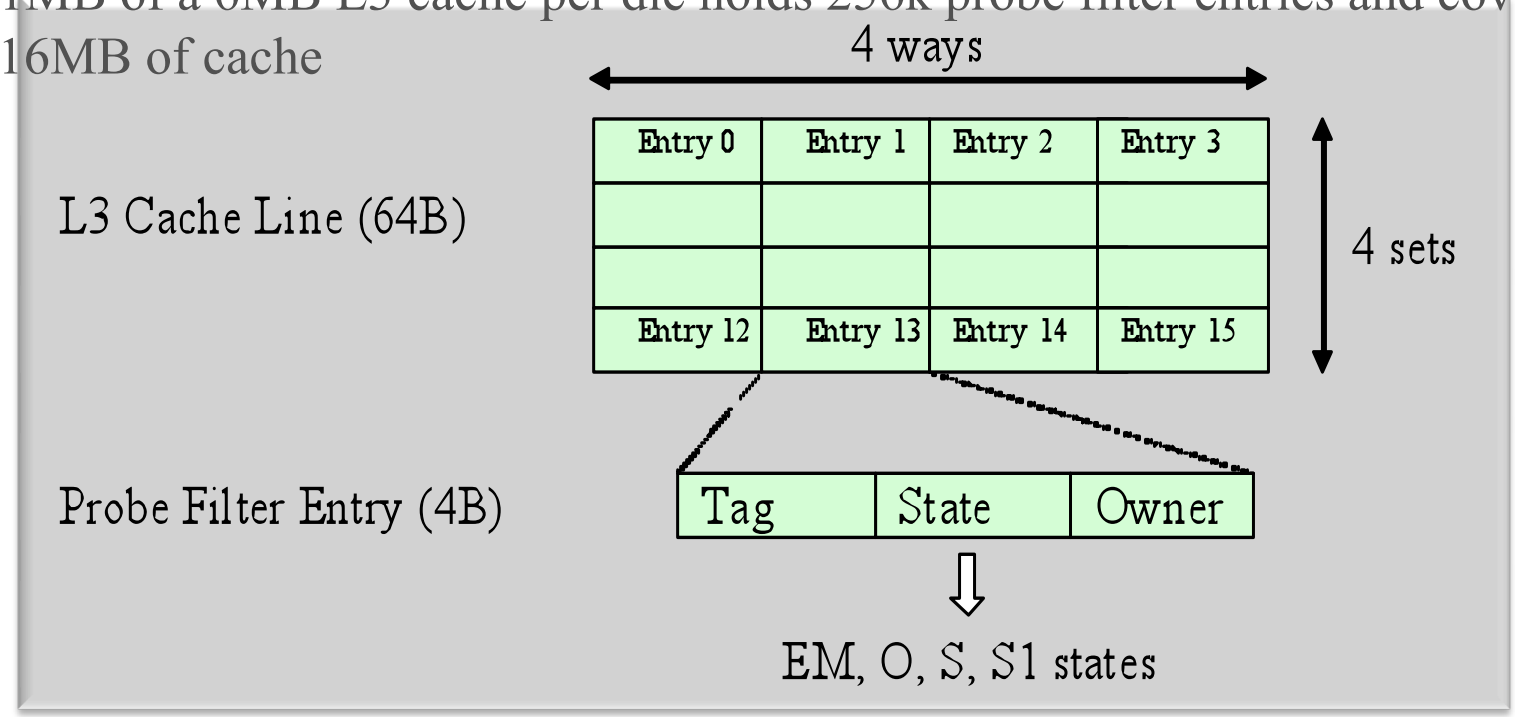
- **A:** Steal 1MB of 6MB L3 cache per die in "Magny-Cours" systems



- Implementation in fast SRAM (L3) minimizes
  - Access latency
  - Port occupancy of read-modify-write operations
  - Indirection latency for cache-to-cache transfers

Format of a Probe Filter Entry

- 16 probe filter entries per L3 cache line (64B), 4B per entry, 4-way set associative
- 1MB of a 6MB L3 cache per die holds 256k probe filter entries and covers 16MB of cache



Software

Compiler Options

- The Portland Group (PGI) family of compilers
  - Support for Linux and Windows.
  - Debuggers and Profilers for OpenMP and MPI.
- GCC and GNU Tools for AMD (gcc, glibc, binutils/gdb)
  - AMD actively contributes improvements targeting AMD cpus.
  - More information available at http://developer.amd.com/cpu/gnu/Pages/default.aspx
- Intel family of compilers
- Pathscale compilers
- Cray compilers

Some Notable PGI Flags

A Good Base Set of Flags

C/C++

-fastsse -Mipa=fast -Mipa=inline -tp shanghai-64

Fortran

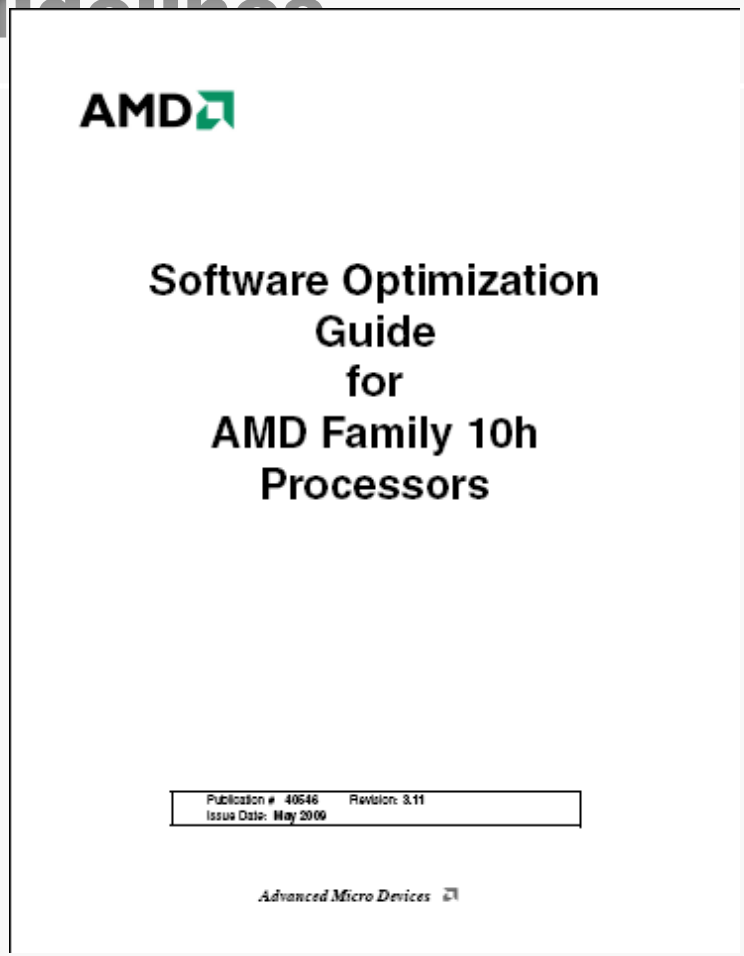
-fastsse -Mvect=short -Mipa=fast -Mipa=inline -tp shanghai-64

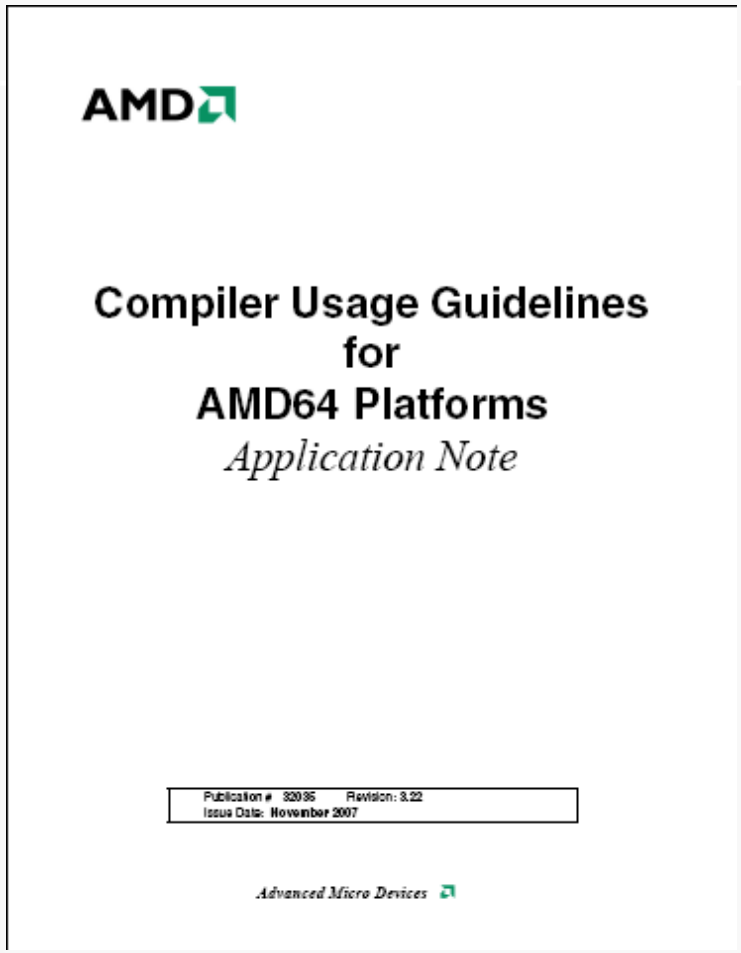
(note: "-tp shanghai-64" is also appropriate for Istanbul)

And Some More to Usually Consider

Flag	Purpose
-Msmartalloc	Use smart malloc routines.
-Msmartalloc=huge	Use smart malloc routines with large pages (depends on amount of OS allocated large pages and the PGI_HUGE_PAGES environment variable).
-Mvect=fuse	Enables vectorizer to fuse loops.
-Mpfi=indirect (first pass) -Mpfo=indirect (second pass)	Use profile feedback optimizations. (requires compiling, doing training run(s), and recompiling).
-Msafeptr (C/C++ only)	Says arrays don't overlap and different pointers point to distinct locations. (has many sub-options).
-Mfprelaxed	Allows relaxed precision for certain Intrinsic functions (sqrt, rsqrt, order, div).

Software Optimization & Compiler Guidelines



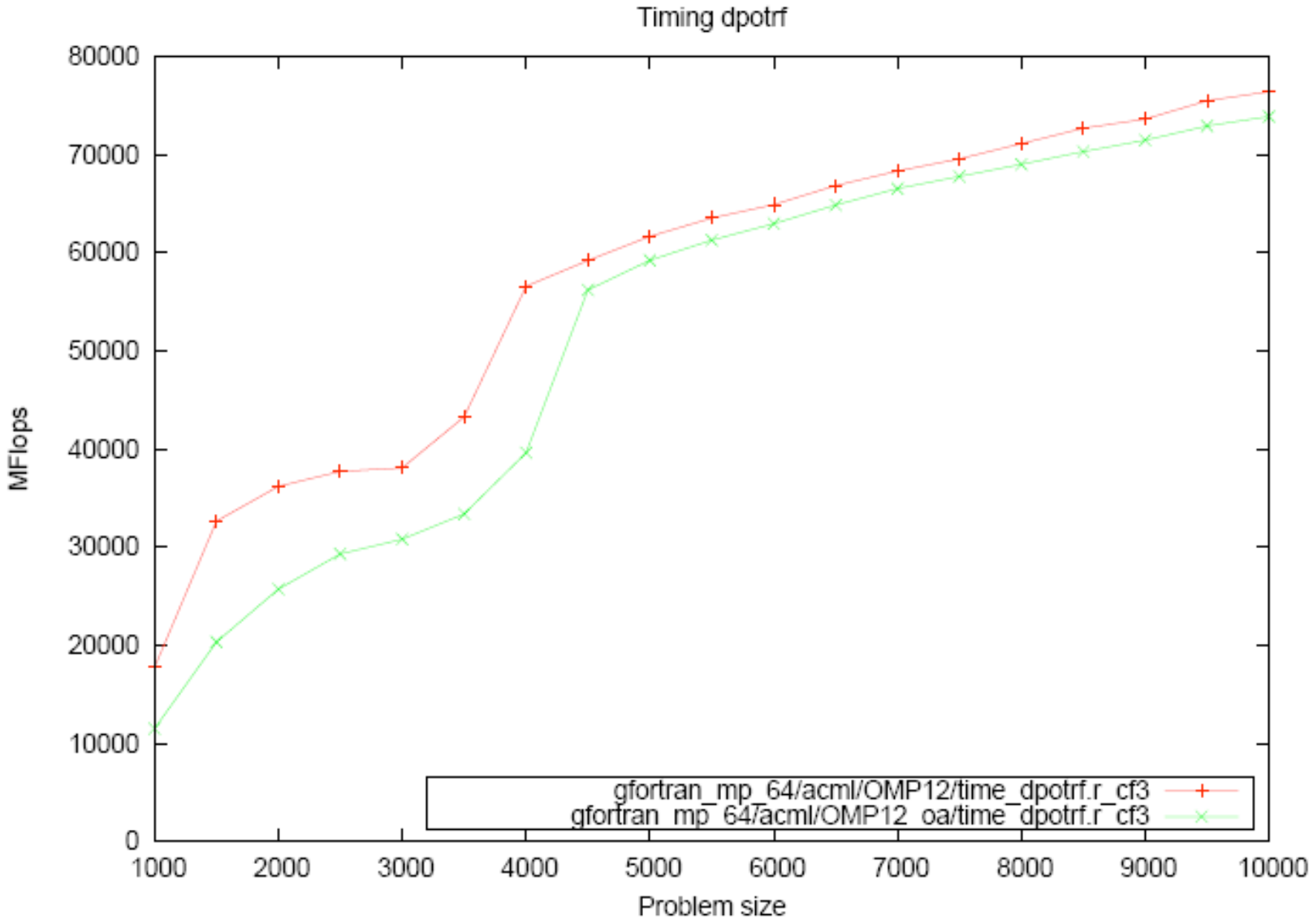


http://developer.amd.com/documentation/guides/Pages/default.aspx

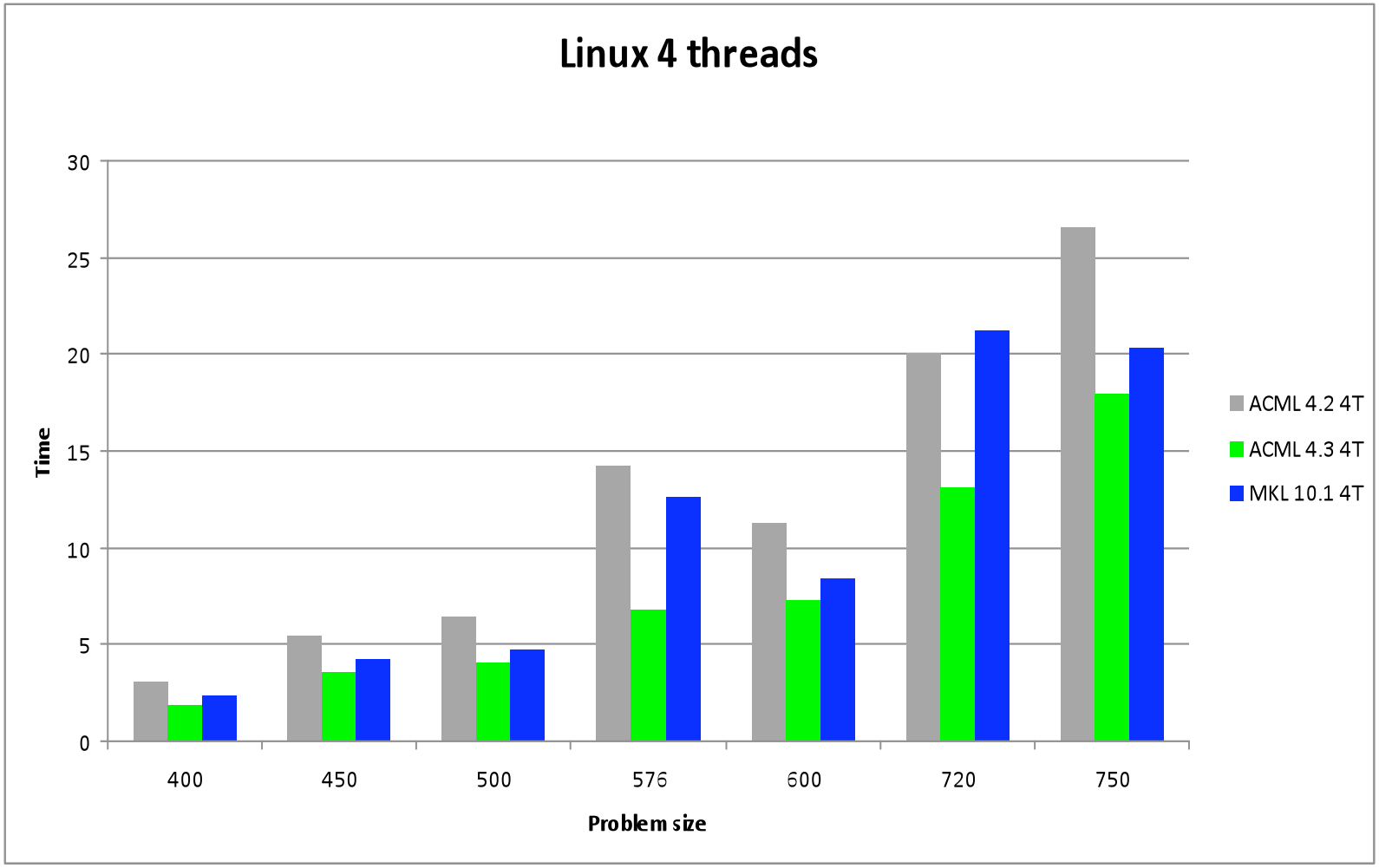
ACML 4.3.0

- L3 BLAS improvements
  - SGEMM for Six-Core AMD Opteron™ Processor
  - New Intel DGEMM and SGEMM kernels
    - Supporting Woodcrest, Penryn, Nehalem
    - Competitive with MKL
  - New DGEMM "fast memory allocation" scheme
    - allows improved performance of other routines (such as LAPACK) which make heavy use of DGEMM
- Six-Core AMD Opteron™ processor tuning for Level 1 BLAS
  - xDOT, xCOPY, xAXPY, and xSCAL
- 3DFFT performance improvements
  - Now outperforming MKL
- AMD Family 10h tuning for real-complex FFTs
  - csfft, dzfft, scfft and zdfft have been re-tuned for FP128

New Memory Allocation Scheme



3DFFT Performance Improvement



Lower is better

ACML 4.3.0 Supported Compilers

- PGI
  - 8.0-6
- GCC/GFORTRAN
  - 4.3.2
  - Now backwards compatible with GCC 4.1.2 and 4.2
- Open64
  - 4.2.1
- Intel Fortran 11.0

ACML 4.4
(Coming Soon)

- Ensure proper scaling with Istanbul/Magny-Cours
- Resolve scaling issues with small problems
- Family 10h optimized ZGEMM, associated L3 routines
- Double Complex 2D/3D-FFT improvements

AMD Developer Central
(for more info and downloads)

- Downloads
  - ACML and AMD LIBM are offered as free downloads to registered members of AMD Developer Central.
- Forums
  - The ACML forum is an excellent place to find answers.
- Blogs
  - Watch for blog entries by members of the ACML and LIBM team.
- **http://developer.amd.com**

Trademark Attribution

AMD, the AMD Arrow logo and combinations thereof are trademarks of Advanced Micro Devices, Inc. in the United States and/or other jurisdictions. Other names used in this presentation are for identification purposes only and may be trademarks of their respective owners.

Questions

References

Backup

MOESI Cache Coherency Protocol